Traffic Light Mapping, Localization, and State Detection for Autonomous Vehicles

Jesse Levinson*, Jake Askeland†, Jennifer Dolson*, Sebastian Thrun*
*Stanford Artificial Intelligence Laboratory, Stanford University, CA 94305
†Volkswagen Group of America Electronics Research Lab, Belmont, CA 94002

Abstract— **Detection of traffic light state is essential for autonomous driving in cities. Currently, the only reliable systems for determining traffic light state information are non-passive proofs of concept, requiring explicit communication between a traffic signal and vehicle. Here, we present a passive camera-based pipeline for traffic light state detection, using (imperfect) vehicle localization and assuming prior knowledge of traffic light location. First, we introduce a convenient technique for mapping traffic light locations from recorded video data using tracking, back-projection, and triangulation. In order to achieve robust real-time detection results in a variety of lighting conditions, we combine several probabilistic stages that explicitly account for the corresponding sources of sensor and data uncertainty. In addition, our approach is the first to account for multiple lights per intersection, which yields superior results by probabilistically combining evidence from all available lights. To evaluate the performance of our method, we present several results across a variety of lighting conditions in a real-world environment. The techniques described here have for the first time enabled our autonomous research vehicle to successfully navigate through traffic-light-controlled intersections in real traffic.**

I. INTRODUCTION

Reliably detecting the state of traffic lights, where $state \in \{red, yellow, green\}$, is essential for autonomous driving in real-world situations. Even in non-autonomous vehicles, traffic light state detection would also be beneficial, alerting inattentive drivers to changing light status and making intersections safer. Non-passive traffic light systems that broadcast their current state and other information have been demonstrated in industry and academic settings [1], and can reliably provide exact state information to vehicles. However, such active systems require expensive hardware changes to both intersections and vehicles, and thus have yet to materialize in any significant market. Thus, in the domain of autonomous driving, reliance on non-passive systems for conveying traffic signal information is not currently feasible, as it requires new infrastructure.

While prior work has demonstrated proof of concept in camera-based traffic light state detection, in recent years the data rate and accuracy of the sensors required for passive traffic light state detection has increased, and supporting sensors such as GPS have improved to the point where such a system could conceivably be operated safely in real-time. The key to safe operation is the ability to handle common failure cases, such as false positives and transient occlusion, which arise frequently and test the limits of camera-only methods.

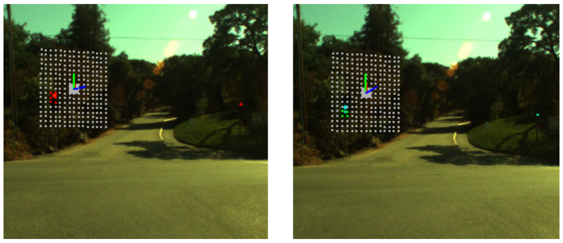


Fig. 1. This figure shows two consecutive camera images overlaid with our detection grid, projected from global coordinates into the image frame. In this visualization, recorded as the light changed from red to green, grid cells most likely to contain the light are colored by their state predictions.

To overcome the limitations of purely vision-based approaches, we take advantage of temporal information, tracking and updating our estimate of the actual light location and state using a histogram filter. To somewhat constrain our traffic light search region, we pre-map traffic light locations and therefore assume that a prior on the global location of traffic lights is available during detection, utilizing pose data available from a GPS system.

Approaching this problem with prior knowledge of the traffic light location has been mentioned in previous work [2], though the implications of such an assumption, including specific methods for dealing with sources of error, have not be thoroughly explored. One contribution of our work is a principled analysis and modeling of the sources of error in each stage of our traffic light state detection, and the creation of a general framework for evidence gathering in a camera and global object prior system.

With respect to our system, global coordinates of each relevant traffic light are quickly obtained, as discussed in Section II during a pre-drive of the course. Such prior knowledge also facilitates the assignment of traffic lights to specific lanes of the road, which is required for efficient route planning and safe driving.

As we do in this paper, several approaches in the literature have focused on camera-based detection of traffic lights. In 1999 [3] described general methods for detection of traffic lights and other urban-driving information, but lacked the sensors and processing power to completely test and deploy their system. In [4] images are analyzed based on color similarity spaces to detect traffic lights, but the method does not run in real-time. Somewhat similar to our image processing algorithm, the detector in [5] biases toward circular regions of high intensity surrounded by regions of low intensity to

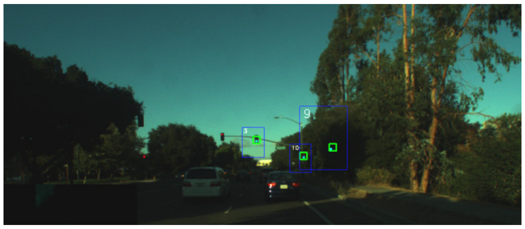


Fig. 2. Simultaneous detection of three distinct traffic lights at one intersection.

represent a light lens inside a darker frame. Their work also attempts to report the state of the light; however, their method does not reliably differentiate between multiple lights, or determine the state of more than one light in a single camera frame.

The experiments of [2] show that traffic light detection is possible in real-time, and that a traffic light detection system can be used in conjunction with a GPS-based navigation system. Although the methods in this work are effective in detecting lights, they do not clearly define a way to avoid false positives in their detection. They also utilize a grid-based probabilistic weighting scheme to estimate the location of a light in the current frame based on its location in the previous frame. Unlike our method, however, they define their grid in image-space, where we define our grid in global coordinates, allowing for motion in image-space due to perspective changes, which may occur as a vehicle approaches a light. The method of [6] uses a three-stage pipeline consisting of a detector, tracker, and classifier, and operates under the assumption that traffic signals may appear anywhere (although they note that this can be alleviated with enhanced maps and GPS information). Using matched filtering and shaped detection based on a Hough Transform, they focus on real-time detection of the lights themselves, as opposed to reliable state detection, showing that the light shapes can be matched in both color and gray scale images. In [7] the focus is also on achieving a high detection rate by thresholding based on intensity information to determine if lights are present in the camera image.

Thus, while employing similar concepts to previous work in some respects, we take a more rigorous probabilistic approach to the entire pipeline which allows for improved handling of several sources of uncertainty. In addition, to our knowledge, ours is the only approach that attempts to detect the state of multiple traffic lights within an intersection in real-time (see Figure 2). As we will show, this insight significantly improves reliability. Finally, we believe we are the first to present quantitative accuracy results for a fixed algorithm operating in several different lighting conditions, which is clearly crucial for a real system.

The remainder of the paper is organized as follows. We discuss traffic light mapping in Section II. In Section III we detail our traffic light state detection pipeline and discuss how we handle sources of uncertainty. We then evaluate an implementation of our system in Section IV.

II. Traffic Light Mapping

To obtain light locations in global coordinates, we drive a route once and record a sensor log comprising both vehicle pose and camera data. Upon review of this log, we manually select lights relevant to our trajectory from video and track them using the algorithm CamShift [8], which attempts to adjust the bounds of an ellipse such that a hue histogram taken over its interior matches that of the original selection.

For each frame in which the light is tracked, the set $X := \{(u,v), C, R\}$ is recorded, where (u,v) are the image coordinates of the ellipse's center pixel and C and R are the estimated global camera position and orientation, respectively. To reduce subtractive cancellation during future calculations, we store the vehicle's global coordinates from the first frame in which the light is selected as a local origin C_0. Then, for each X, we find the ray $d = (a,b,c)$ from the camera lens with local position $C - C_0 = (x,y,z)$ to the traffic light lens using the back projection formula:

$$d = \lambda R^{-1} K^{-1} (u,v,1)^T, \tag{1}$$

where K is the camera's intrinsic parameter matrix and λ normalizes d to unit length. Once light tracking has finished, the optimal point of intersection from each ray is determined as in [9] from the following. Suppose the light is tracked for n frames, let

$$A = \begin{pmatrix} \sum_i^n (1-a_i^2) & -\sum_i^n a_i b_i & -\sum_i^n a_i c_i \\ -\sum_i^n a_i b_i & \sum_i^n (1-b_i^2) & -\sum_i^n b_i c_i \\ -\sum_i^n a_i c_i & -\sum_i^n b_i c_i & \sum_i^n (1-c_i^2) \end{pmatrix} \tag{2}$$

$$b = \begin{pmatrix} \sum_i^n [(1-a_i^2)x_i - a_i b_i y_i - a_i c_i z_i] \\ \sum_i^n [-a_i b_i x_i + (1-b_i^2) y_i - b_i c_i z_i] \\ \sum_i^n [-a_i c_i x_i - b_i c_i x_i + (1-c_i^2) z_i] \end{pmatrix} \tag{3}$$

then the estimated global light position is given by $l_{est} = A^{-1} b + C_0$. The resulting traffic light locations are stored in a text file which is read by the traffic light detection system described in Section III. Also during tracking, a bitmap of the ellipse's interior is stored at five meter intervals of vehicle travel for use in our probabilistic template matching algorithm (see Section III-D).

III. Traffic Light State Detection

A. Prominent Failure Cases

Our system must be robust to pathological situations, like superfluous lights and temporary occlusions, and must also work at all times of day, under all possible lighting conditions.

Assuming that a vision algorithm could distinguish tail lights from traffic lights during the day in real-time, the task becomes impossible at night with most video cameras. All distinguishing features, aside from light color, are lost. Since in the United States there is no legally-defined standard for light lens material and intensity [10], tail lights near where we expect a traffic light could be considered traffic lights in the image processing algorithm.

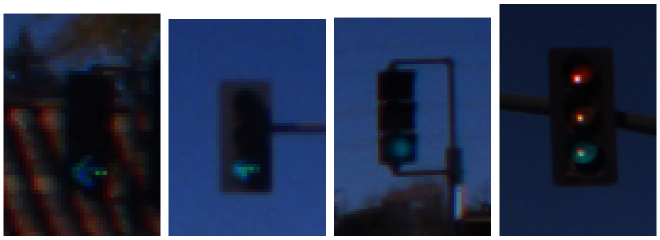


Fig. 3. Many traffic lights appear almost as dim as their surroundings either by design or by accident.

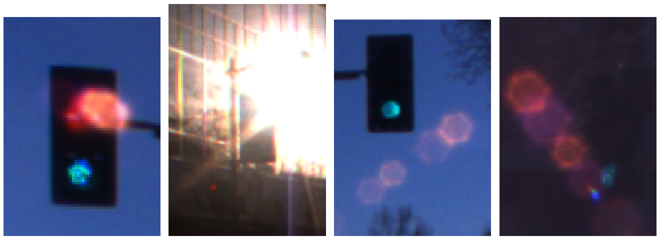


Fig. 4. Often, a lens flare or back lighting will obscure a light's state. Using cheaper consumer-grade cameras could necessitate additional noise filtering and perhaps explicit flare detection.

In the following section, we will describe the probabilistic techniques used to make our system invariant to lighting conditions and reliable in situations that would be challenging for a vision-only approach.

Two major technical problems have to be solved in order to be able to detect the state of a traffic light robustly:

1) Inferring the image region which corresponds to the traffic light.
2) Inferring its state by analyzing the acquired intensity pattern.

Ideally, in solving these problems we would like to choose a reference frame that allows us to take advantage of temporal consistency. The choice of our detection grid as a reference frame assumes several structured error components (as we discuss below), allowing the light's position within the grid to change slowly over time. Given this temporal constraint, and our vision algorithm that performs within the limits of reasonable expectations, we can then apply a histogram filter to infer the image region of the light and determine the color, as discussed in the sections that follow.

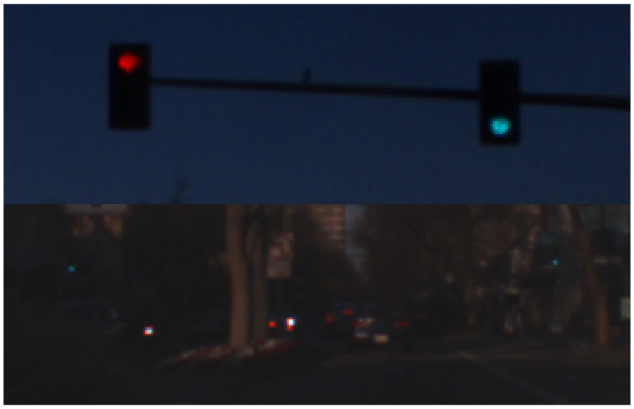


Fig. 5. (Top) A left-pointing red arrow appears in the camera image as circular, making non-contextual state detection difficult. (Bottom) The vehicle needs to know the state of the next intersection far in advance so appropriate actions are taken. In this case, 200mm green lights appear far apart among a cluttered scene with many light sources.

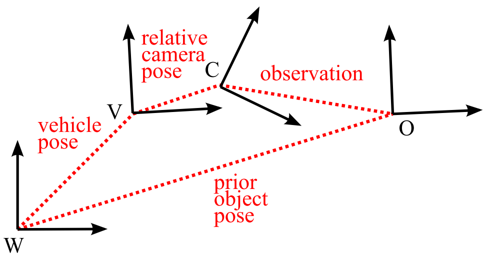


Fig. 6. Several random variables and their uncertainties influence the accuracy of the localization result. This diagram shows the dependencies of the different coordinate frames: [W]orld, [V]ehicle, [C]amera and [O]bject (here: traffic light).

B. Traffic Light Tracking

Consider the diagram in Figure 6. Two chains of information lead to the observation of an object in the camera image. Given the distances at which traffic lights have to be detected (70-80m for our vehicle to be able to react appropriately at reasonable traveling speeds), small errors in camera calibration or vehicle localization can lead to large deviations between the expected location of the traffic light on the image and the actual one. From our experience, this error lies in the order of 50 pixels (corresponding to 5-10m in scene space) for state-of-the-art localization hardware and algorithms [11], [12].

Current computer vision algorithms are not yet capable of reliable, holistic scene understanding such that the small image of the traffic light (with the lens less than 3-5 pixels wide) could be distinguished reliably from surrounding patterns.

We propose to make explicit the mismatch between expected perceptions and actual ones (hereafter called perceptual offset) by introducing a random time-varying vector $o^t = (o_1^t, o_2^t)$. Expected perceptions include the position predicted by the mapping, localization, and GPS components of the pipeline while actual perceptions are the data scores returned by our image processing algorithm. We update our belief about o^t sequentially using a Bayes filter, implemented as a histogram filter. One of the critical questions that arises is in which coordinate frame the perceptual offset is represented best. We argue that two straight-forward choices are suboptimal. First, tracking the displacement in the image coordinate frame leads to drastically changing posteriors at every movement of the vehicle. This introduces an unnecessary estimation bias, since the actual displacement effect varies only slowly. Second, tracking the offset in complete three dimensional world space, that is, essentially treating the world pose of the traffic light as a random variable and updating it over time, is hard to accomplish robustly, due to the nature of the construction of a camera image, our main data source. Without a planar constraint, all positions along the ray from the camera sensor through the plane could be weighted equally, since they all project to the same pixel on the image plane.

In contrast to these choices, we found that the most suitable space for modeling the perceptual offset is a bounded plane centered and oriented as the traffic light in world space. Our grid is a natural way to restrict the implied distance from the camera, and is supported by our assumption of a fairly good light mapping prior with respect to latitude and longitude. We represent our belief about the offset at time t by a histogram over this bounded plane (i.e., a normalized grid) and update it recursively according to Bayes rule [13]

$$P(o^t : z^t, o^{t-1}) = \nu \cdot P(z^t : o^t) \cdot P(o^t : o^{t-1}). \qquad (4)$$

We assume a relatively peaked Gaussian motion model for $P(o^t : o^{t-1})$ to account for changes to the perceptual offset caused by motion of the vehicle and the vehicle pose uncertainties, and mapping uncertainties (observed as light movement in the detection grid frame due to camera perceptive changes not properly captured) implicit in this motion, where the standard deviation of the Gaussian is proportional to $\varepsilon + k$, where $k \propto \textit{vehicle speed}$. The observation model $P(z^t : o^t)$, which defines the relationship between camera observations z^t and the perceptual offset o^t of the traffic light is detailed further below.

The prior distribution of o^t on the grid is a two dimensional Gaussian distribution centered at the mapped traffic light location with standard deviations chosen according to the expected error in mapped traffic light location plus the expected perceptual offset. From our experience, these quantities are easy to optimize using a recorded training sequence of traffic light observations.

Our approach works under the following assumptions:

1) The maximal perceptual offset (depending on the localization accuracy of the vehicle and the camera calibration) is smaller than half the side lengths of the grid used to track it.
2) The resolution of the grid is high enough such that traffic light observations falling onto the border of two grid cells are not represented as two distinct lights with disparate locations (as light location is estimated at the grid cell center).
3) The camera image plane remains approximately parallel to the grid plane during the tracking process. We found that an orientation mismatch of up to about 35 degrees does not pose a problem (and this is already beyond they legally possible orientation mismatch between traffic lights and their corresponding lanes).
4) Neighboring traffic lights are spaced at least half the side length of the grid apart from one another. If this is not the case, the lights should be modeled jointly, which can be achieved in a straightforward way by constructing a joint sensor model with additional color channels, though this is not further addressed here.

When making decisions about the state of the traffic light, the current pose uncertainty of the vehicle - that is, its current posterior distribution over poses - has to be considered. This can be achieved by convolving the current state of the histogram filter for the vehicle pose with the current state of the histogram filter for the perceptual offset, taking the geometric transform between the two grid planes into account.

Given the posterior estimate of the perceptual offset (represented by a histogram) over grid cells, we select the cell containing the mode as the most likely location of the light. We then report the most likely state at that grid cell, as determined by our data scores.

C. Uncertainty Discussion

In our system, the quality of input sources varies. Regarding the map input *latitude*, *longitude*, *altitude*, and *orientation* for each light, we assume a greater accuracy in the first two coordinates than in the last. Without a good estimate of *latitude* and *longitude* determining which lane a traffic light applies to would be nearly impossible. This accuracy requirement is also supported by our method for collecting light location data.

Our light mapping procedure depends on accurate GPS and camera calibration. In our system, even when the car is driving directly towards the light, this method typically gives repeatable errors which are easily compensated for with a static offset. The result is a light localized to within approximately two meters across and four meters high.

Another source of uncertainty is the projection of points from global coordinates back to a given pixel in the camera image. To project a three dimensional coordinate in some object space into the two dimensional image plane of a camera, first one usually transforms the coordinate so that its origin is the camera center. To perform this transform beginning with $(\textit{latitude}, \textit{longitude}, \textit{altitude})$, we first transform our coordinates to the projected Universal Transverse Mercator (UTM) coordinates, then to "smooth coordinates", which are calculated by integrating our GPS-reported velocity over time. This coordinate system is "smooth" because it does not suffer from the abrupt changes of raw GPS information, but does drift over time. We then transform smooth coordinates to the vehicle's local frame, where the origin moves with the vehicle. After this we transform our coordinates to the camera frame. Uncertainty is captured in our random variable C in camera calibration and extrinsic camera parameters, both in terms of static errors and those due to motion or vibration of the vehicle, as well as camera sensor noise.

We used a Flea imager by Point Grey, set to fixed gain, shutter speed and saturation such that the intensity and saturation of the center pixel on an illuminated, 200mm, LED-based green traffic light at 45 meters were reported as 99% and 92%, respectively. The saturation of the sky at noon on a clear day was reported as approximately 66%. Because we are detecting light sources, fixing our camera parameters implies the intensity and saturation of these light sources, as our camera observes them, will remain constant with respect to environmental lighting.

Even with fixed camera parameters, lens type and particular color characteristics of a light are probable culprits for false negatives as discussed in Section III-A and so are an important set of uncertainty sources. Our image processing algorithm handles these variations by building hue and saturation histograms from the lenses tracked during the light mapping routine. These histograms are then used as lookup tables (henceforth referred to as Histogram As Lookup-Tables or HALTs) which condition our images before template matching occurs.

D. Probabilistic Template Matching

The constraints of the traffic light state detection problem are favorable for using template matching to generate higher-order information from raw camera data. The color of an active lens, since it emits light, remains nearly constant regardless of external lighting conditions. Traffic lights in particular are distinguishable from most other light sources by their strong color saturation. However, variation among lenses and minor effects due to external light direct us to use color probabilistically.

Bitmaps of traffic light lenses $\{\mathbf{B}_k\}_{k=1...n}$ are captured during a pre-drive. From these bitmaps, hue and saturation images $\{\mathbf{h}_k\}_{k=1...n}$ and $\{\mathbf{s}_k\}_{k=1...n}$ are extracted. For each state $\omega \in \{red, yellow, green\}$, HALTs H_ω (Figure 7) and S_ω, which represent histograms of lens hue and saturation, are generated from (5) and (6). Lookup-table V, representing a distribution of the expected intensity value of pixels lying on the traffic light's black frame, is defined by (7). We denote the height and width of the k^{th} image h_k and w_k.

$$H_\omega[x] = \sum_{k=1}^{n}\sum_{i=1}^{h_k}\sum_{j=1}^{w_k} \delta(\mathbf{h}_{k,i,j}, x), \qquad x = 0...359 \tag{5}$$

$$S_\omega[x] = \sum_{k=1}^{n}\sum_{i=1}^{h_k}\sum_{j=1}^{w_k} \delta(\mathbf{s}_{k,i,j}, x), \qquad x = 0...255 \tag{6}$$

$$V[x] = 255 \cdot \exp(-0.035x), \qquad x = 0...255 \tag{7}$$

$$\delta(a,b) = \begin{cases} 0 & \text{for } a \neq b \\ 1 & \text{otherwise.} \end{cases} \tag{8}$$

In California, traffic light lenses have radii of either 200 or 300 mm [10]. A len's resulting radius in pixels can be approximated from the camera's intrinsic parameters magnification m and focal length f, the camera lens to light source lens distance D and the physical radius of the traffic light lens R, as in (9).

$$r = \left\lfloor \frac{2Rmf}{D} \right\rfloor \tag{9}$$

The adaptively generated light lens template $\mathbf{T}(r)$ is modeled as a circle of diameter $2r+1$ pixels centered over a black square with side length $l = 4r + (1 - (4r \mod 2))$ after a linear convolution (denoted $\otimes$) with a 3x3 Gaussian kernel $G(\sigma)$ with $\sigma = 0.95$ [14].

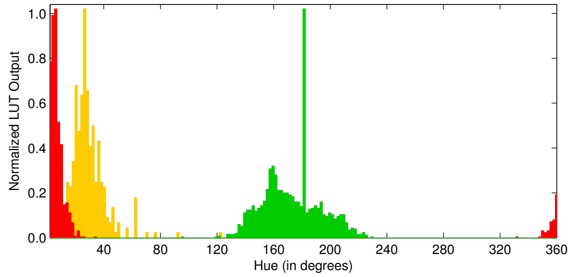


Fig. 7. Hue histograms H_ω are generated from individual lights at various distances, captured during a pre-drive.

$$\hat{\mathbf{T}}_{i,j}(r) = \begin{cases} 0 & \text{if } ||(i-2r+1, j-2r+1)|| > r \\ 255 & \text{otherwise,} \end{cases} \tag{10}$$

for $i, j = 1...l$

$$\mathbf{T}(r) = G(\sigma) \otimes \hat{\mathbf{T}}(r) \tag{11}$$

In the image processing pipeline, we limit ourselves to the regions of interest projected onto our image plane from the detection grid (ergo, Figure 1). For the remainder of this section, we will consider each such region as an independent image $\mathbf{I}$. Image $\mathbf{I}$ is split into hue, saturation and value images $\mathbf{H}$, $\mathbf{S}$ and $\mathbf{V}$. For an image $\mathbf{N}$, we denote its height $h_{\mathbf{N}}$ and width $w_{\mathbf{N}}$. As a preprocessing step, we heavily weight saturated and high intensity pixels in $\mathbf{S}$ and $\mathbf{V}$ by applying transforms N_S and N_V.

$$N_S(x) = x^5/2^{32}, \; x \in \{0...255\} \tag{12}$$

$$N_V(x) = \begin{cases} 1 & \text{for } x <= 60 \\ x & \text{otherwise.} \end{cases}, \; x \in \{0...255\} \tag{13}$$

$$\mathbf{U}_{i,j} = N_S(\mathbf{S}_{i,j}) N_V(\mathbf{V}_{i,j}) \tag{14}$$

Image $\mathbf{G}_\omega$ is generated as follows:

$$\mathbf{G}_\omega(\mathbf{H},\mathbf{S},\mathbf{V}) = \sum_{i=1}^{h_{\mathbf{I}}}\sum_{j=1}^{w_{\mathbf{I}}} \left\{ \mathbf{U}_{i,j} \sum_{k=1}^{h_{\mathbf{T}(r)}} \sum_{l=1}^{w_{\mathbf{T}(r)}} \left(L(\mathbf{H},\mathbf{S}) + F^2(\mathbf{V}) \right) \right\} \tag{15}$$

$$L(\mathbf{H},\mathbf{S}) = \frac{H_\omega[\mathbf{H}_{u(k,l)}] S_\omega[\mathbf{S}_{u(k,l)}] (1 - \delta(\mathbf{T}_{k,l}(r), 0))}{\max\{\mathbf{T}(r)\} - \mathbf{T}_{k,l}(r) + 1} \tag{16}$$

$$F(\mathbf{V}) = V[\mathbf{V}_{u(k,l)}] \delta(\mathbf{T}_{k,l}(r), 0), \tag{17}$$

where $u(k,l) = (i + k - h_{\mathbf{T}}/2, j + l - w_{\mathbf{T}}/2)$. To eliminate non-circular regions of bright saturation, we then apply a linear convolution with circle template $\mathbf{C}$ over $\mathbf{G}_\omega$:

$$\mathbf{C}_{i,j} = \begin{cases} 0 & \text{if } 2r <= ||(i-2r+1, j-2r+1)|| \\ -1 & \text{if } r <= ||(i-2r+1, j-2r+1)|| < 2r \\ 1 & \text{otherwise,} \end{cases} \tag{18}$$

for $i, j = 1...l$

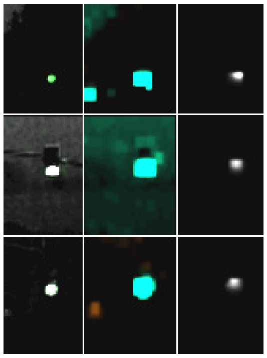


Fig. 8. (Left) is a visualization constructed by merging $\{Q_{red}, Q_{green}, Q_{yellow}\}$ as the *red*, *green* and *blue* channels of an image. Color saturation corresponds to a higher certainty of a specific state in the current frame (before the prior probability is processed). (Center) depicts the detection grid image D, in which each cell's color represents a grid cell's predicted state and its intensity represents that state's score. (Right) is the resulting prior as described in Section III-B. Each light captured in figure 2 is shown from top to bottom.

$$\mathbf{Q}_{\omega,i,j} = \max\left(\{\mathbf{C} \otimes \mathbf{G}_{\omega}\}_{i,j}, 0\right) \qquad (19)$$

Since the projection of our detection grid $\mathbf{D}$ onto the image plane does not guarantee square regions, we use a convex polygon filling algorithm – quad-filling – as discussed in [15]. For grid cell $\mathbf{D}_{i,j}$, we project the region in $\mathbf{Q}_{\omega}$ that falls on $\mathbf{D}_{i,j}$'s interior. We then select the maximum value in this region to be this cell's score $\mathbf{E}_{\omega,i,j}$.

E. State Detection Pipeline

For a single light:

1) Supply the module with a pre-defined list of global traffic light locations, specified by *latitude*, *longitude* and *altitude*.
2) Supply the module with histograms of hue and saturation for regional traffic lights.
3) Begin looking for lights at a distance determined by the camera resolution and vehicle braking distance.
4) Generate a grid, in global coordinates, centered on the expected location of the traffic light.
5) Project the grid $\mathbf{D}$ onto the camera image plane to determine region of interest image $\mathbf{I}$.
6) Compute $\mathbf{Q}_{\omega}$ for each $\omega \in \{red, yellow, green\}$.
7) Back-project $\mathbf{Q}_{\omega}$ onto $\mathbf{D}$ to get score image $\mathbf{E}_{\omega}$.
8) Set $P(state) := (G(\sigma_s) \otimes P(state))\mathbf{E}_{\omega}^2$

For multiple lights per intersection:

Given a state $\omega \in \{red, yellow, green\}$, an intersection I with n lights, and a traffic light l_k, $k \in \{1 \ldots n\}$ with histogram filter outputs $P(l_k = \omega)$ at a particular intersection, the probability for the intersection state is given by

$$P(I = \omega) = \frac{\prod_k P(l_k = \omega)}{\sum_{c \in \{r,y,g\}} \prod_k P(l_k = c)} \qquad (20)$$

which treats each light as an independent measurement of the true intersection state, and so the final decision is given by $max_j\{P(I = c_j)\} \Rightarrow I = c_i$.

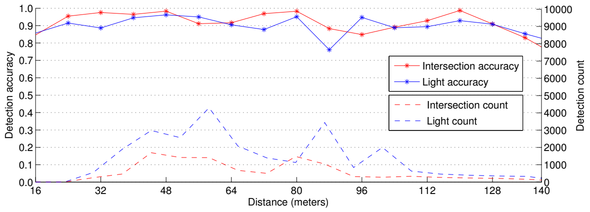


Fig. 9. Correct detection rates of individual lights and complete intersections from noon are depicted.

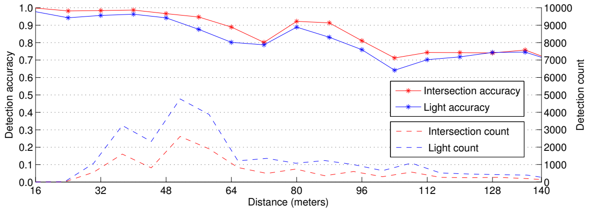


Fig. 10. Correct detection rates of individual lights and complete intersections from sunset are depicted.

IV. EVALUATION AND RESULTS

We demonstrate the success of our techniques on Junior, Stanford University's autonomous research vehicle. Junior is equipped with an Applanix LV-420 tightly coupled GPS/IMU system that provides inertial updates and global position estimates at 200 Hz. Although we typically run an additional layer of laser-based localization to refine the GPS pose[16], for the sake of more generally applicable results we disable this refinement localization and rely only on the default GPS output in the following results. Traffic lights are seen by a Point Gray "Flea" video camera which provides 1.3 megapixel RGB frames at 15 Hz.

To demonstrate robustness in various traffic light scenarios, and under multiple lighting conditions, we recorded a 20-minute route through Palo Alto, California at each of three

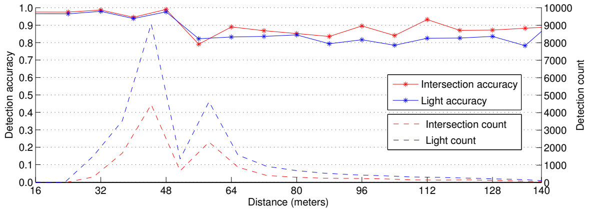


Fig. 11. Correct detection rates of individual lights and complete intersections from night are depicted.

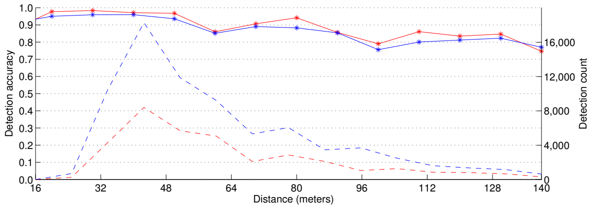


Fig. 12. Combined results from noon, sunset and night are depicted.

different times: noon, sunset, and night. All parameters were fixed across all times of day, indicating that the exact same algorithm is effective across a wide range of lighting conditions. The route comprised 82 lights across 31 intersections, some of which were extremely challenging (e.g. Figure 5, bottom).

Figures 9-11 depict correct detection rates of individual lights and complete intersections under the three lighting conditions, as a function of distance to the lights. Figure 12 aggregates all three runs and shows our overall results. Unsurprisingly, detection accuracy drops off somewhat with distance, as the sizes of the traffic lights in the image decrease.

Comparison of individual light detections and intersection decisions suggests a distinct advantage and improved robustness with the latter, when multiple traffic lights are known to display the same color at an intersection. Indeed, using the Bayesian approach as described previously, we are frequently able to correctly identify the true state of the intersection's color even when confronted with conflicting individual detections. As Figure 13 shows in two separate examples, the probabilistic approach often yields the desired posterior even in extremely difficult scenes where one or more lights is incorrectly classified.

Mathematically, as long as the sources of error in each simultaneous detection are not fully dependent, a combined approach will be superior in expectation. Indeed, in Figure 14 we see that our algorithm's confidence in its detection label is strongly correlated with its accuracy, so that two or more lights which disagree will often be disambiguated correctly based on the detection(s) which are more confident.

A confusion matrix for individual light detections across all three times of day is shown in Figure 15. Similarly, a confusion matrix for intersection decisions is shown in Figure 16. Again, we see that intersections are meaningfully more accurate than individual lights. These results also indicate that we are very accurate at detecting red lights, and less accurate at detecting green and especially yellow lights. Fortunately for safety purposes, we are much more likely to mistakenly label a light as red than as any other color.

We calculate accuracy based on the fraction of frames containing a correct intersection state classification, out of the total number of video frames in our approximately 20-minute test sequence over three sequences for which intersections' lights are visible. Of the 76,310 individual light detections across 82 lights each at three times of day, we achieve 91.7% percent accuracy. For the associated 35,710 intersection decisions across 31 intersections, again at three times of day, we achieve 94.0% accuracy.

	Noon	Sunset	Night	Combined
Lights	92.2%	88.9%	93.8%	91.7%
Intersections	95.0%	92.3%	95.0%	94.0%

A simple extension to the standard framewise approach that is useful for practical applications is to only report traffic light state when several frames of identical color detections

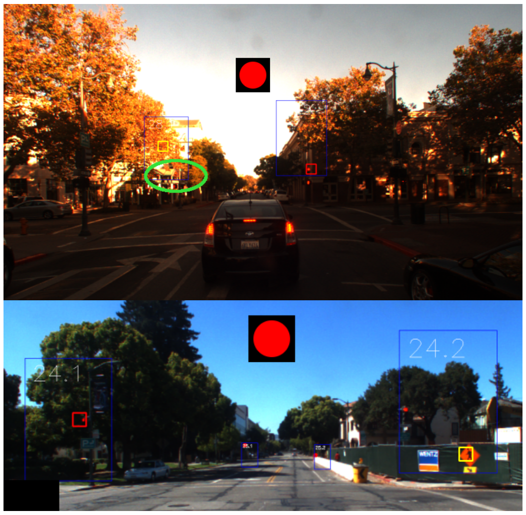


Fig. 13. Considering multiple lights with separate search windows (here, shown in blue) improves robustness. In the top frame, the window on the left reports a yellow light with 38% probability and the window on the right reports a red light with 53% probability. The final state decision of the intersection, taking into account probabilities for red, yellow and green states from both windows, yields (correctly) a red state with probability 53%. A green ellipse surrounds the actual red light that we missed, illustrating how difficult the problem is. In the bottom frame, probabilities are 80% and 48% for the left and right lights and 87% for the intersection, whose state is correctly detected as red.

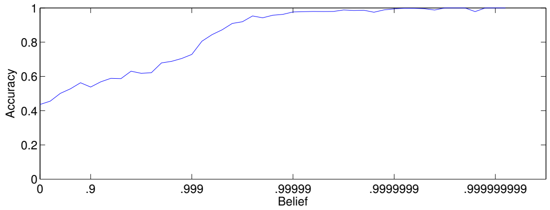
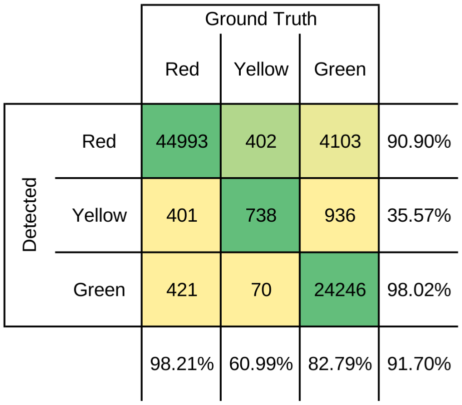


Fig. 14. Our belief that a given light has a particular state versus the accuracy of that belief, using raw output from histogram filters. Although the filter is generally overconfident, its certainty does correlate strongly with accuracy, as desired.



		Ground Truth			
		Red	Yellow	Green	
Detected	Red	44993	402	4103	90.90%
	Yellow	401	738	936	35.57%
	Green	421	70	24246	98.02%
		98.21%	60.99%	82.79%	91.70%

Fig. 15. Confusion matrix of light detections. Entries are number of detections. Each row and column has an associated percent accuracy. Yellow lights are more easily confused due to their shared hues with red lights and because it is more difficult to obtain a large training set for relatively rare yellow lights as compared to red and green lights.

		Ground Truth			
		Red	Yellow	Green	
Detected	Red	21015	178	1505	92.59%
	Yellow	41	341	232	55.54%
	Green	142	25	12231	98.65%
		99.14%	62.68%	87.56%	94.05%

Fig. 16. Confusion matrix of intersection decisions. Entries are number of decisions. Each row and column has an associated percent accuracy.

have occurred sequentially. Although this extension adds a fraction of a second of latency, the response time still matches or exceeds that of a human, and the results are several (absolute) percentage points higher than the framewise results above. Thus for use in an autonomous vehicle, this application of hysteresis is preferred.

In addition to these quantitative results, a significant implication of this work is that Junior is now able to autonomously drive through intersections governed by traffic lights. However, a safety driver is always present to ensure correct behavior, as we have certainly not solved traffic light detection as well as humans. In addition, if we avoid autonomous driving at intersections whose lights are especially difficult to detect, and utilize our localization system which reduces uncertainty in traffic light location, our actual performance is significantly better than the preceding results might suggest.

V. Conclusions and Future Work

We have described a novel pipeline for traffic light state detection. We take a principled approach to collection of higher-level information from our camera image, utilizing strong constraints in template creation and weighting. Accounting for possible sources of error inherent in the traffic light detection problem, we specifically analyze those errors which contribute to uncertainty in our pipeline. And, for the first time, we have shown that the detection of multiple lights for each intersection improves robustness to noise and significantly improves performance relative to the single-light case.

There remains progress to be made on this challenging topic. Certainly, a higher resolution camera system would improve traffic light clarity at longer distances; currently, at the longer detection ranges, traffic lights often only occupy one or two pixels. An interesting extension would be to combine the camera vision system with 3-dimensional LIDAR data in order to explicitly detect traffic lights by shape in addition to color. On the other side of the cost spectrum, cheaper consumer-grade cameras could be deployed, which would necessitate additional noise filtering and perhaps explicit flare detection (see Figure 4). Finally, overall results may be improved if our traffic light detection algorithms were probabilistically combined with computer vision algorithms for holistic scene detection, as false positives may be more easily suppressed.

VI. Acknowledgments

We are deeply grateful to Christian Plagemann for valuable advice and discussion, to Mike Sokolsky for maintaining the autonomous vehicle research platform and assisting with data collection, and to Jan Aue and Florian Schuller for meticulous data labeling. Jennifer Dolson acknowledges support from an NDSEG Graduate Fellowship from the United States Department of Defense.

References

[1] V. Gradinescu, C. Gorgorin, R. Diaconescu, V. Cristea, and L. Iftode, "Adaptive traffic lights using car-to-car communication." in *VTC Spring*. IEEE, 2007, pp. 21–25. [Online]. Available: http://dblp.uni-trier.de/db/conf/vtc/vtc2007s.html

[2] T.-H. Hwang, I.-H. Joo, and S. I. Cho, "Detection of traffic lights for vision-based car navigation system," in *PSIVT*, 2006, pp. 682–691.

[3] U. Franke, D. Gavrila, S. Görzig, F. Lindner, F. Paetzold, and C. Wöhler, "Autonomous driving goes downtown," *IEEE Intelligent Systems*, vol. 13, no. 6, pp. 40–48, 1998.

[4] T. Shioyama, H. Wu, N. Nakamura, and S. Kitawaki, "Measurement of the length of pedestrian crossings and detection of traffic lights from image data," *Measurement Science and Technology*, vol. 13, no. 9, pp. 1450–1457, 2002. [Online]. Available: http://stacks.iop.org/0957-0233/13/1450

[5] J.-H. Park and C. sung Jeong, "Real-time signal light detection," in *FGCNS '08: Proceedings of the 2008 Second International Conference on Future Generation Communication and Networking Symposia*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 139–142.

[6] F. Lindner, U. Kressel, and S. Kaelberer, "Robust recognition of traffic signals," in *Intelligent Vehicles Symposium, 2004 IEEE*, June 2004, pp. 49–53.

[7] S.-K. Joo, Y. Kim, S. I. Cho, K. Choi, and K. Lee, "Traffic light detection using rotated principal component analysis for video-based car navigation system," *IEICE Transactions*, vol. 91-D, no. 12, pp. 2884–2887, 2008.

[8] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *Information Theory, IEEE Transactions on*, vol. 21, no. 1, pp. 32–40, January 2003.

[9] G. Slabaugh, R. Schafer, and M. Livingston, "Optimal ray intersection for computing 3d points from n-view correspondences," pp. 1–11, 2001.

[10] C. D. of Transportation, *California Manual on Uniform Traffic Control Devices*, September 2006.

[11] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun, "Junior: The stanford entry in the urban challenge," *Journal of Field Robotics*, 2008.

[12] J. Levinson, M. Montemerlo, and S. Thrun, "Map-based precision vehicle localization in urban environments," in *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.

[13] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello, "Bayesian filtering for location estimation," *IEEE Pervasive Computing*, vol. 2, pp. 24–33, 2003.

[14] G. Bradski and A. Kaehler, *Learning OpenCV*, 1st ed. O'Reilly Media, Inc., 9 2008, p. 112.

[15] J. F. S. Hill and S. M. Kelley, *Computer Graphics Using OpenGL (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.

[16] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, May 2010, pp. 4372–4378.